CRADA Final Report
for
CRADAs ORNL 96-0431 and 99-0564

Intelligent Machine Learning Analysis For Fuel Cell Operations

R. W. Murphy
Oak Ridge National Laboratory
Oak Ridge, Tennessee

and

W. A. Hoyt
ERC, Inc.
Huntsville, Alabama

June 30, 2000

PROTECTED CRADA INFORMATION

List of Figures

INTELLIGENT MACHINE LEARNING ANALYSIS FOR FUEL CELL OPERATIONS: SOLID OXIDE FUEL CELL GENERATOR SYSTEM MODEL

ABSTRACT

A performance computational model for a 100 kW nominal solid oxide fuel cell generator system is described. The calculational methods are based on the FORTRAN programming language. Comprehensive parameter input options are presented, and constraints are identified. Example reactant, electrical, and efficiency outputs are demonstrated over the relevant operating ranges. A sample calculated output display at nominal operating conditions is given.

BACKGROUND

The relatively high operating temperatures (600-1100°C) of solid oxide fuel cells offer the promises of internal reforming (elimination of external reformer component) of natural gas fuel, rapid kinetics (without the involvement of precious materials), immunity to "poisons" (carbon monoxide actually serves as a fuel), and high quality product heat for bottoming cycle and/or valuable process use, but offer substantial challenges regarding construction materials and control requirements (Ref. 1). Performance models can assist in exercising and evaluating selected control options.

MODEL DEVELOPMENT

A FORTRAN language-based model of performance for a 100 kW nominal solid oxide fuel cell generator system (illustrated in Fig. 1) was developed. The model characterizes setpoints, controls, and outputs of the system for selected fuels, operational states, and operator inputs within identified envelope limits. The main program accesses system limit information stored in an ASCII file named LIMITS.DAT and operator input information stored in a separate ASCII file named OPINPUT.DAT to avoid the necessity of recompiling the main FORTRAN file each time a change in limit or operator input parameters is desired. A description of the details of the model is presented below. Figure 2 shows a model flow chart for convenient reference. The remaining figures show sample relationships between parameters calculated by the model and operating currents for selected values of other operator inputs.

INPUTS

Cell Stack and Geometry

The generator module is comprised of a "stack" or array of 1152 individual tubular solid oxide fuel cells as shown in Fig. 3. They are arranged as 12 strings (of 96 cells each) series-connected in a serpentine configuration with one air-cooled power lead at each end for direct-current power takeoff. Each string consists of 2 segments (of 48 cells each) also connected in series. Each segment consists of 2 bundles (of 24 cells each) again connected in series. As the fundamental building block of the module, a bundle has 3 parallel current paths (circuits) formed from 3 rows of 8 series-connected cells each. Each cell has a length of 150 cm and an outside diameter of 2.22 cm. The module arrangement provides fuel to the outside of each tubular cell and air to the inside. Thus the outside electrode serves as the anode by producing electrons for the external power connection while the inside electrode serves as the cathode by accepting electrons from the external power connection. The oxygen ions are conducted through each cell's solid oxide electrolyte with an effective conduction area of 810 cm^2.

Fuel Characteristics

The primary fuel supply for the generator system is natural gas. The composition of this fuel is subject to change both as a function of time (for example, summer low demand versus winter peaking) and location [for example, the Westinghouse (PNG/WESTC) location in the United

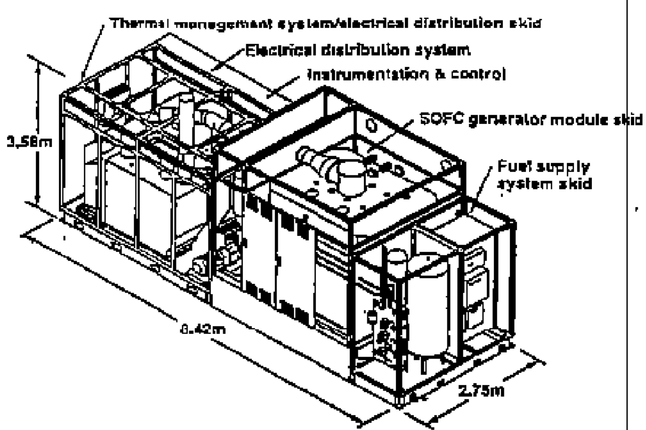

Fig. 1. Isometric Drawing of the 100 kW Power System.

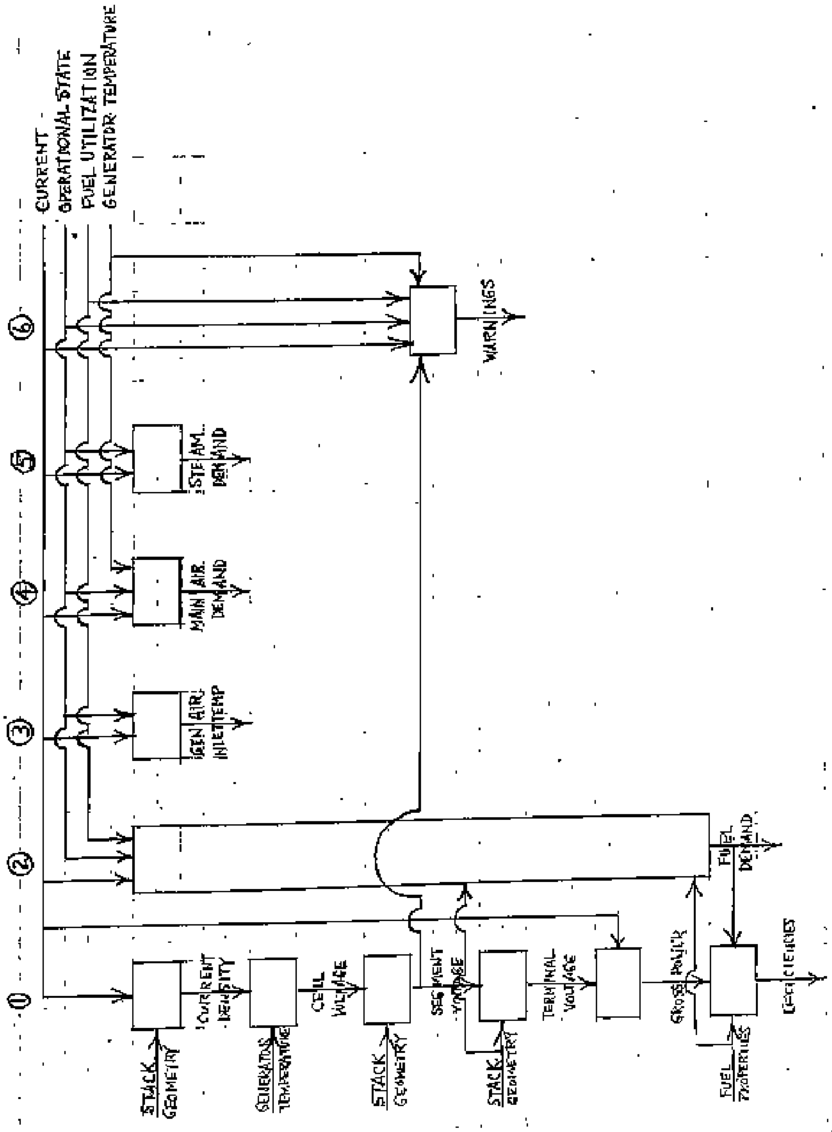

Fig. 2. Flow Chart for 100 kW Solid Oxide Fuel Cell Generator System Model.

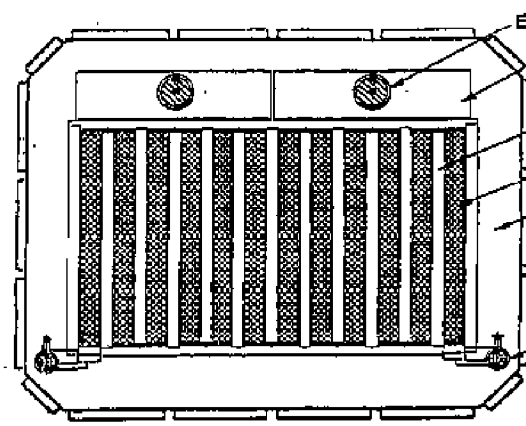

Fig. 3. Layout of the 100 kW Stack.

States versus the Nuon (EDB/ELSAM) location in the Netherlands]. In principle, the solid oxide fuel cell is capable of operation with hydrogen, hydrocarbon, or carbon monoxide fuel. However, with natural gas, the main constituent of the fuel is methane, with small fractions of higher hydrocarbons (ethane, propane, butane, etc.) also being present. For present purposes, mixed hydrocarbon fuel is characterized by the average number of moles of reformed fuel (hydrogen and carbon monoxide) produced from each mole of unreformed fuel. If the unreformed fuel contained only methane, the relevant value would be 4. For the PNG/WESTC location, this value is taken to be 4.056. For the EDB/ELSAM location, the corresponding value is 3.5606. Related fuel properties include effective molecular weight and heating values (lower and higher).

Generator Temperature

The generator temperature is an operator input intended to achieve and maintain the cell electrolyte at a temperature that can provide the required oxygen ion conductivity, subject to other constraints. At steady state, this temperature is the result of achieving a balance between the heat produced within the generator and the heat removed from it. In practice, control is accomplished by varying bypass air flow around one low and one high temperature recuperator and/or by varying electrical power input to one on/off air heater (100 kW) and one modulating air heater (100 kW max). Normally, heater power is not required when the generator is operating above the "thermal sustain boundary," estimated to occur at a generator current of approximately 407 A. The generator temperature also determines whether or not specific operational states are permitted. For present purposes, cell performance data was available in Ref. 2 only for an operating temperature of 1000°C.

Fuel Utilization

The fuel utilization increment is an operator input intended to control the excess amount of fuel (above that required stoichiometrically for a given current) supplied to the generator. The baseline fuel utilization has been established by the manufacturer to be 85%. At an earlier point, the allowed increment range about this baseline was –3% to +5%, giving a net fuel utilization range of 82% to 90%. Later changes increased the allowed increment range to –8% to +5%, giving a net fuel utilization range of 77% to 90%. For operation in the RUN operational state, fuel utilization is one of the parameters used in the algorithm for determine the desired fuel flow setting.

Current

The current is an operator input that acts as the primary parameter for determining fuel flow, steam flow, air flow, and air inlet temperature setpoints for the generator. Together with the cell stack geometry, the current determines the current density, which, in turn, establishes the cell, segment, and terminal voltages of the stack, as well as, ultimately, the gross stack (direct current) power.

Operational State

The operational state is an operator input that establishes, within constraints, the generator functional mode. The valid states are PRE-OP, HEAT, LOAD, RUN, STOP, COOL, SAFE-STOP. Allowable operation in a particular state or change from one state to another is predicated on satisfaction of selected constraints concerning generator temperature, current, segment voltage, etc. In the model, detailed consideration is limited to the LOAD and RUN states because they are the only normal functional modes in which the generator is producing electrical power.

ALGORITHMS, MODELING, AND OUTPUTS

Current Density, Voltages, and Gross Power

As indicated in the first vertical leg of Fig. 2, first the effective generator current density is determined from the current and the cell stack geometry. Next the cell voltage is determined from the calculated current density and the generator temperature, based on the cell voltage/current density characteristic applicable to the particular generator temperature (Fig. 4 based on Ref. 2). The decrease of cell voltage with increasing current density reflects polarization and other losses characteristic of this type of fuel cell (Ref.1). The corresponding segment (Fig. 5) and terminal (Fig. 6) voltages are calculated in the model from the cell voltage and the cell stack geometry. Established minimum segment voltages are shown in Fig. 5 and checked as indicated in the section on warnings. Established RUN and LOAD state current limits are shown in Figs. 5 and 6 and also checked as indicated in the section on warnings. Next the gross stack power (Fig. 7) is calculated from the terminal voltage and the current. The nonlinearity of the line in Fig. 7 again reflects characteristic losses.

Fuel Flow and Nozzle Pressure

As shown in the second vertical leg of Fig. 2, the fuel demand (Figs. 8-9) and the corresponding nozzle pressure (Figs. 10-11) are calculated from algorithms using fuel utilization, operational state, and current operator inputs, as well as cell stack geometry information. The first leg of the model then uses this result along with fuel properties and gross power to calculate lower (Fig. 12) and higher heating value (Fig. 13) gross electric efficiencies.

The model first calculates the reformed (hydrogen or carbon monoxide) fuel flow in standard liters per minute required to support a current of 1 A generated by one cell. This value is then multiplied by the total generator current and divided by the total number of parallel circuits (presently 3) to determine the reformed fuel flow for one cell at its effective operating current. To this amount a fixed correction for each cell is added (from Ref. 2, currently set at 46 standard cubic centimeters per minute; earlier values ranged from 18 to 21 standard cubic centimeters per minute) to account for apparent leakage effects. This combined amount is then multiplied by the total number of cells in the array to determined the total reformed fuel flow required by the generator at the operating current. This result is then divided by the appropriate average number of moles of hydrogen and carbon monoxide produced from each mole of unreformed fuel

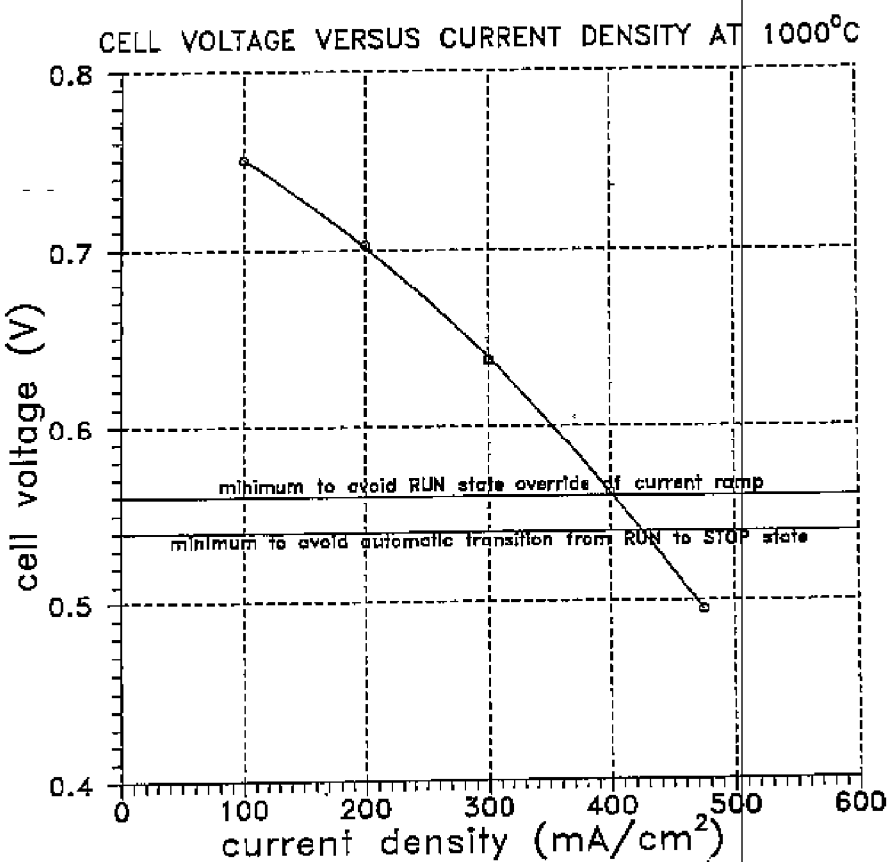

Fig. 4. Cell Voltage Versus Current Density at 1000°C.

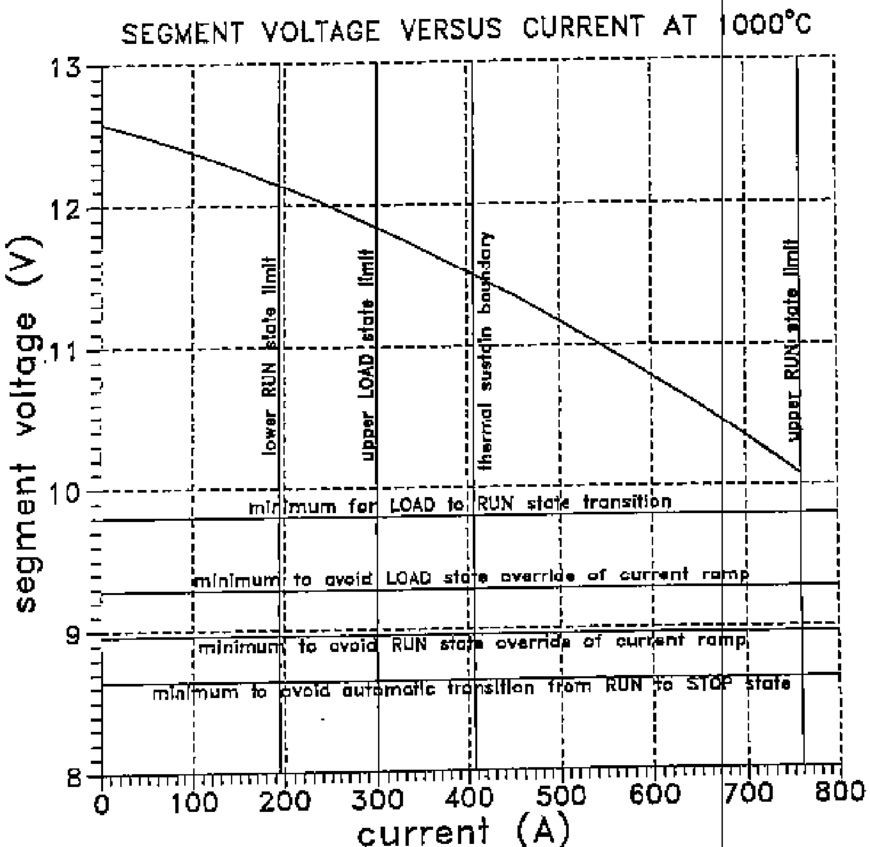

Fig. 5. Segment Voltage Versus Current at 1000°C.

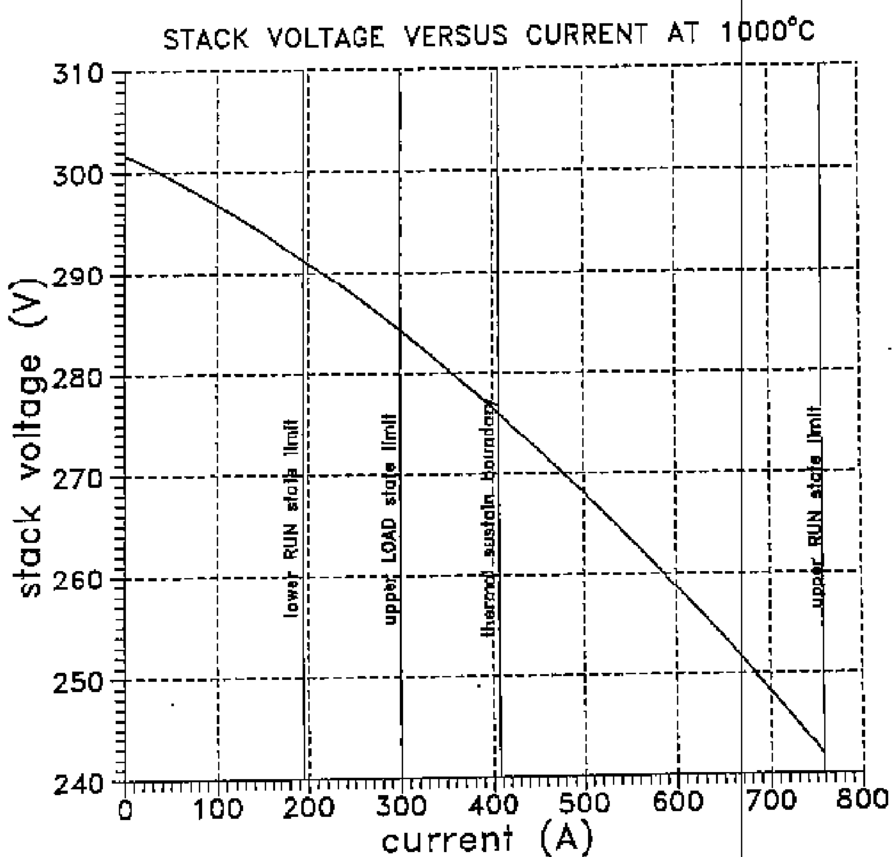

Fig. 6. Stack Voltage Versus Current at 1000°C.

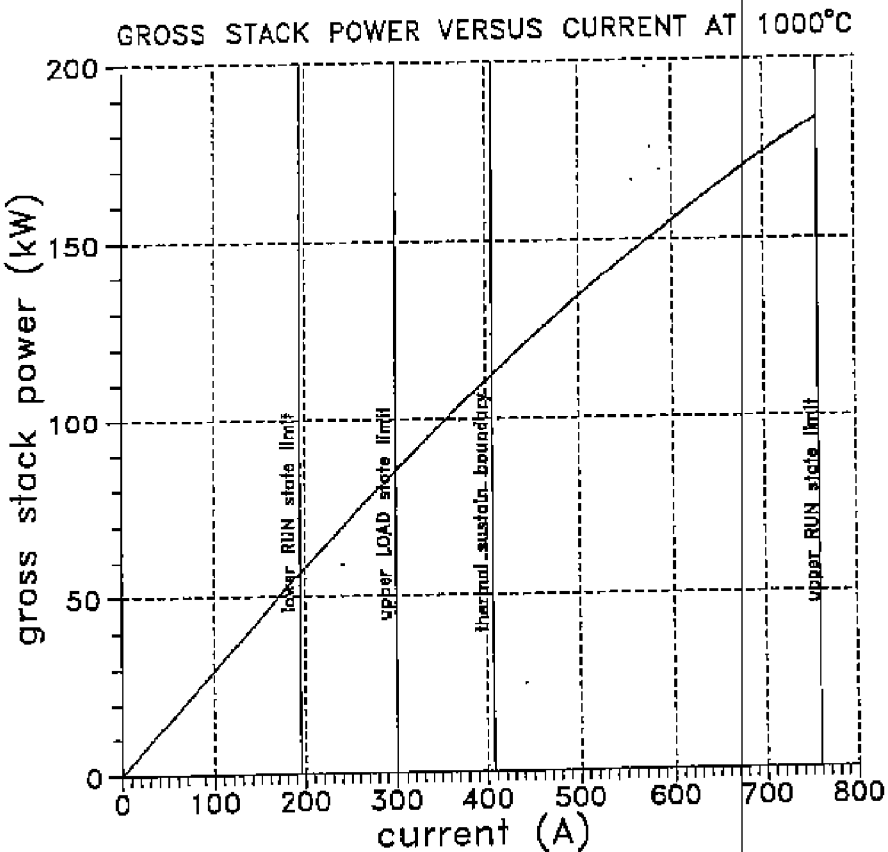

Fig. 7. Gross Stack Power Versus Current at 1000°C.

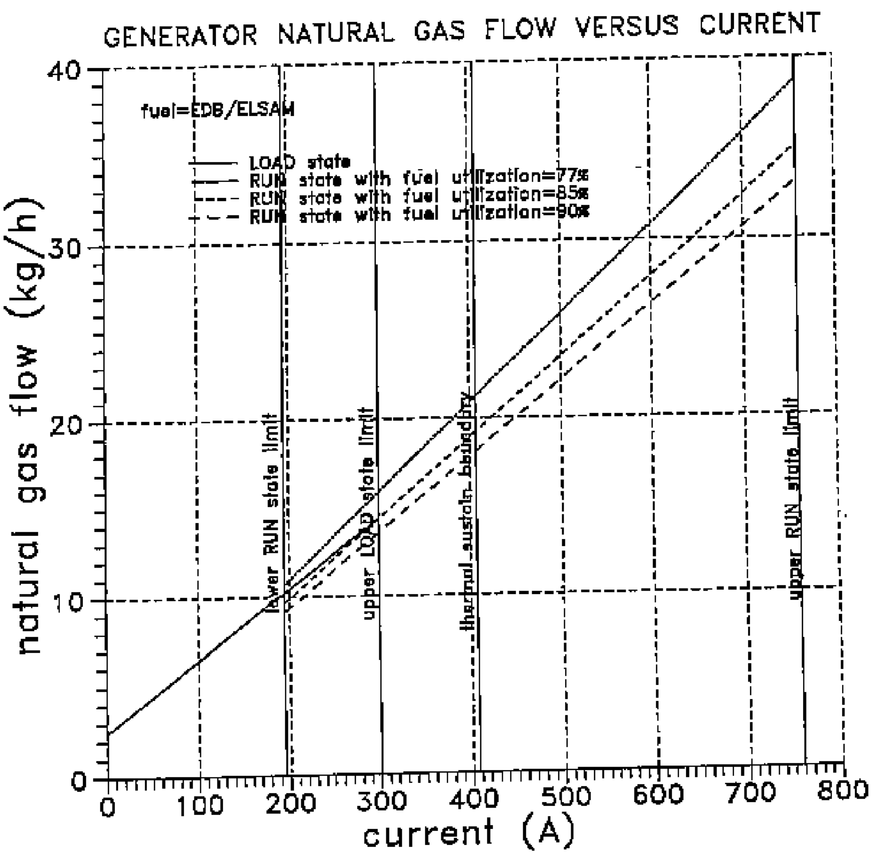

Fig. 8. Generator Natural Gas Flow Versus Current (EDB/ELSAM).

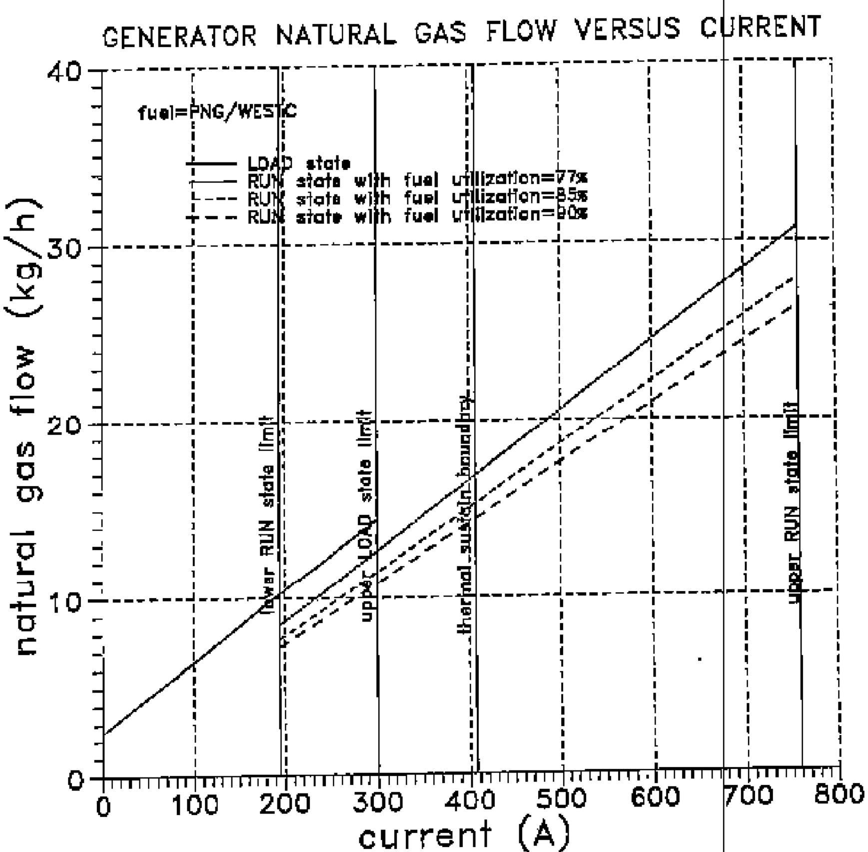

Fig. 9. Generator Natural Gas Flow Versus Current (PNG/WESTC).

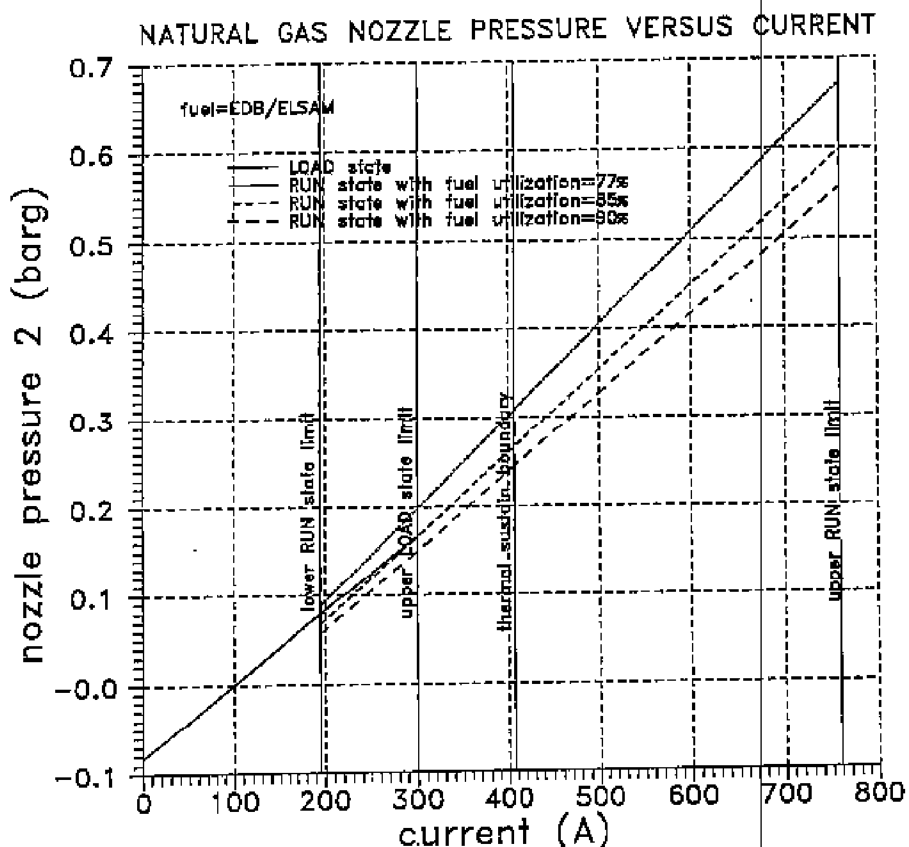

Fig. 10. Natural Gas Nozzle Pressure Versus Current (EDB/ELSAM).

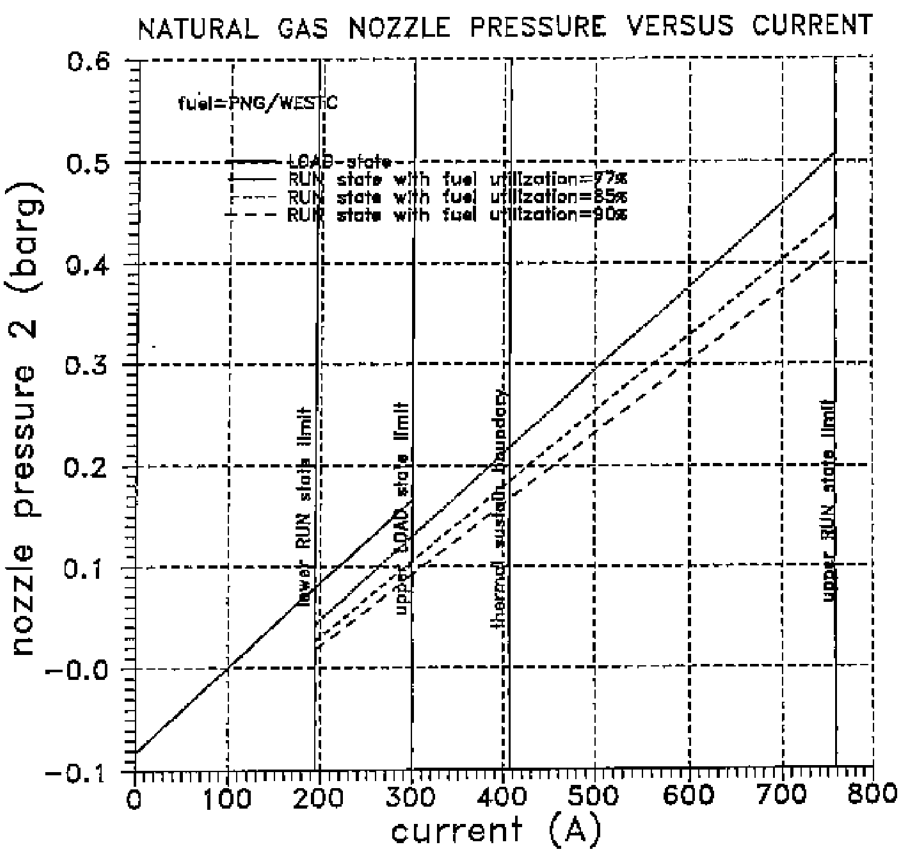

Fig. 11. Natural Gas Nozzle Pressure Versus Current (PNG/WESTC).

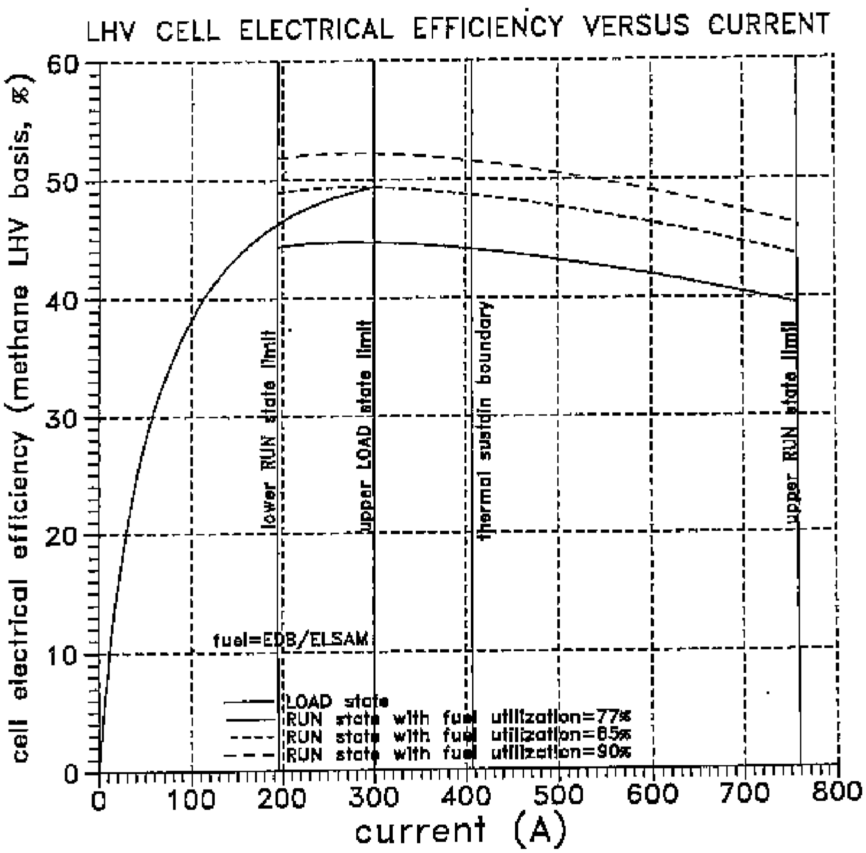

Fig. 12. LHV Cell Electrical Efficiency Versus Current.

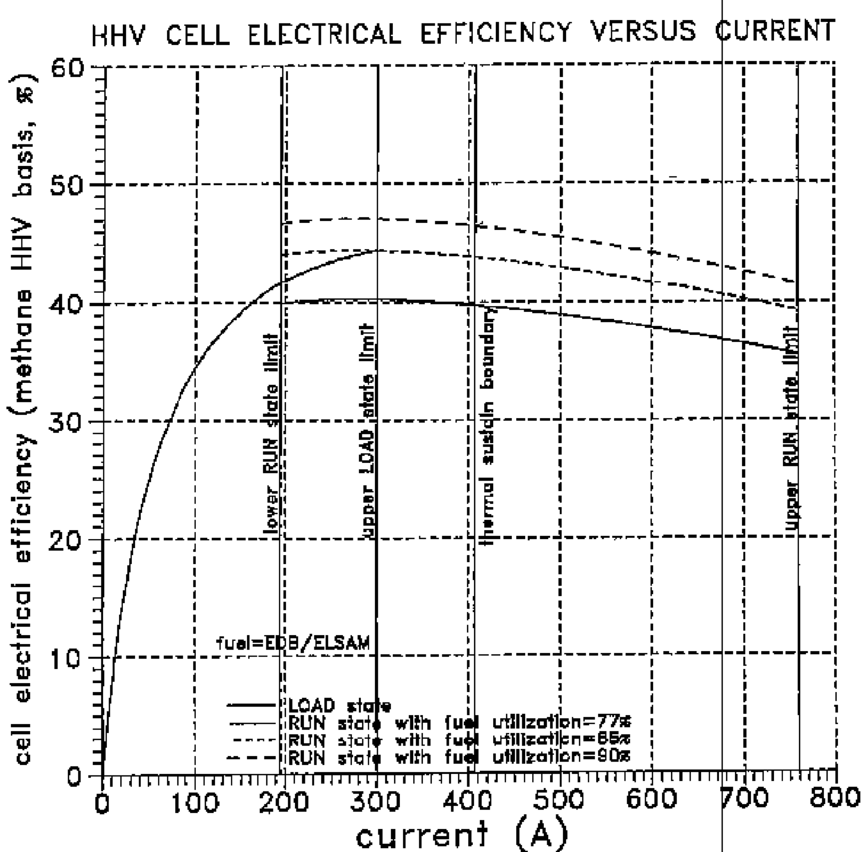

Fig. 13. HHV Cell Electrical Efficiency Versus Current.

(natural gas) to give the unreformed fuel flow in standard liters per minute required if the generator operated at 100% fuel utilization. This value is divided by the fuel utilization and a correction factor (Ref. 2) to give the required unreformed fuel flow in standard liters per minute. Then from the unreformed fuel molecular weight, the related standard density in grams per standard liter is calculated. The previous result is then multiplied by this result to give the required unreformed fuel flow in grams per minute, which is then converted to the desired setpoint units of kilograms per hour.

For the LOAD operational state, the algorithm is a similar but slightly different linear relation between fuel setpoint and current.

Figure 8 shows the results for EDB/ELSAM fuel in both RUN and LOAD operational states over the full projected current operating range. As shown in both Fig. 8 and the LIMITS.DAT file listing, allowable currents are presently limited to values less than 300 A in the LOAD operational state. For the RUN operational state, allowable currents presently range from 195 A to 759 A. The implied dependence of RUN state setpoint values on fuel utilization is also shown using the base, minimum, and maximum levels. Of course, higher fuel utilization corresponds to lower fuel flow for a given current.

Figure 9 shows related results for PNG/WESTC fuel in the same operational states and current range. The LOAD state characteristic is identical to that in Fig. 8 because the relevant algorithm depends only on the operating current. However, the RUN state characteristics are lower than the corresponding ones in Fig. 8 because of its algorithm's dependence on the particular fuel composition. Specifically, referring to the parameters supplied earlier, one mole of unreformed PNG/WESTC fuel is calculated to provide 4.056 total moles of hydrogen and carbon monoxide when reformed, while one mole of unreformed EDB/ELSAM fuel will provide only 3.5606.

For present purposes, the bypass flow around the primary nozzle of the exhaust recirculation ejector is closed. In this situation, all the fuel flows through the primary nozzle, and the model employs an algorithm to relate the primary nozzle pressure to the fuel flow. This relationship, with nozzle pressure expressed in bar gauge units is illustrated in Fig. 10 for EDB/ELSAM fuel and in Fig. 11 for the PNG/WESTC fuel.

To calculate the gross electric efficiency of the generator, the model calculates the ratio of the gross stack power determined earlier to the effective heating value of the fuel supplied to the generator. For the EDB/ELSAM fuel composition, the lower heating value efficiencies are presented in Fig. 12 and the higher heating value efficiencies in Fig. 13 for both LOAD and RUN operational modes and for RUN fuel utilization value corresponding to the base, minimum, and maximum levels. The use of the lower heating value basis gives higher apparent efficiencies because it does not account for the heating value associated with condensing water vapor formed during the reaction process. Also, higher fuel utilization implies less fuel flow for a given power, resulting in higher efficiency. The apparent efficiency maxima indicated in the moderate current range are the result of a combination of inherent loss mechanisms. As current increases, the polarization and other losses, illustrated by the voltage decrease with increasing current density in Fig. 4, reduce the efficiency. However, as current (and, therefore, fuel flow) decreases, the relative importance of the fixed fuel leakage loss increases, thus decreasing efficiency.

Air Inlet Temperature

As indicated in the third vertical leg of Fig. 2, the generator inlet air temperature setpoint is calculated from algorithms employing operational state and current operator inputs. In the LOAD operational state, the generator inlet air temperature setpoint is fixed at 735°C. The same is true for the RUN operational state when the operating current is less than or equal to 300 A. As currents increase above 300 A in the RUN operational state, the generator inlet air temperature setpoint decreases, as shown in Fig. 14, to approximately 444°C at the 759 A current limit to accommodate, in part, the greater heat generation from the stack.

Process Air Supply

As shown in the fourth vertical leg of Fig. 2, the main air demand setpoint (Fig. 15) is calculated from algorithms using generator temperature, operational state, and current operator inputs. In the LOAD operational state, with no current the main air blower demand is set at the HEAT operational fixed value of 88.0%. As current increases in this state, the main air blower demand decreases linearly until it reaches 72.0% at an operating current of 300 A, the upper LOAD state current limit as shown in Fig. 15. Within the RUN operational state, the algorithm incorporates a proportional integral derivative (PID) control component to assist in controlling the generator temperature. The value of the PID output employed can vary between 0.00 and 1.00. With the output value at 0.00, the base algorithm blower demand will be adjusted downward by 10.0%. With the output value at 1.00, the base algorithm blower demand will be adjusted upward by 10.0%. With the output value at 0.50, there will be no adjustment to the base algorithm blower demand. As an option to automatic variation based on generator temperature, the PID output value can be manually set as an operator input. For the RUN operational state, Fig. 15 illustrates the base (PID output =0.50) algorithm's linear decrease in main air blower demand for currents less than 300 A, consistent with that already discussed for the LOAD state. As the operating currents increase above 300 A, the main air blower demand (using the base algorithm) increases linearly from 72.0% at 300 A to 86.9% at the 759 A current limit shown in Fig. 15. Corresponding RUN operational state boundaries established by minimum and maximum allowed values of the PID output are also given in the same figure.

Steam Demand

As shown in the fifth vertical leg of Fig. 2, the steam demand setpoint (Fig. 16) is calculated from algorithms employing operational state and current operator inputs. For present purposes, there is steam demand only when the generator is in the LOAD operational state. Although some steam may be generated by the steam supply system for checkout purposes in other states, the steam demand requirement is primarily for "startup" purposes to provide sufficient water for the internal fuel reforming processes at low loads (currents less than approximately 250 A) during ramp-up. At higher loads, the generator provides adequate quantities of water as a result of its inherent electrochemical processes. As indicated in Fig. 16, for operating currents less than or equal to 150 A in the LOAD operational state, the steam demand is fixed at 80%. For greater

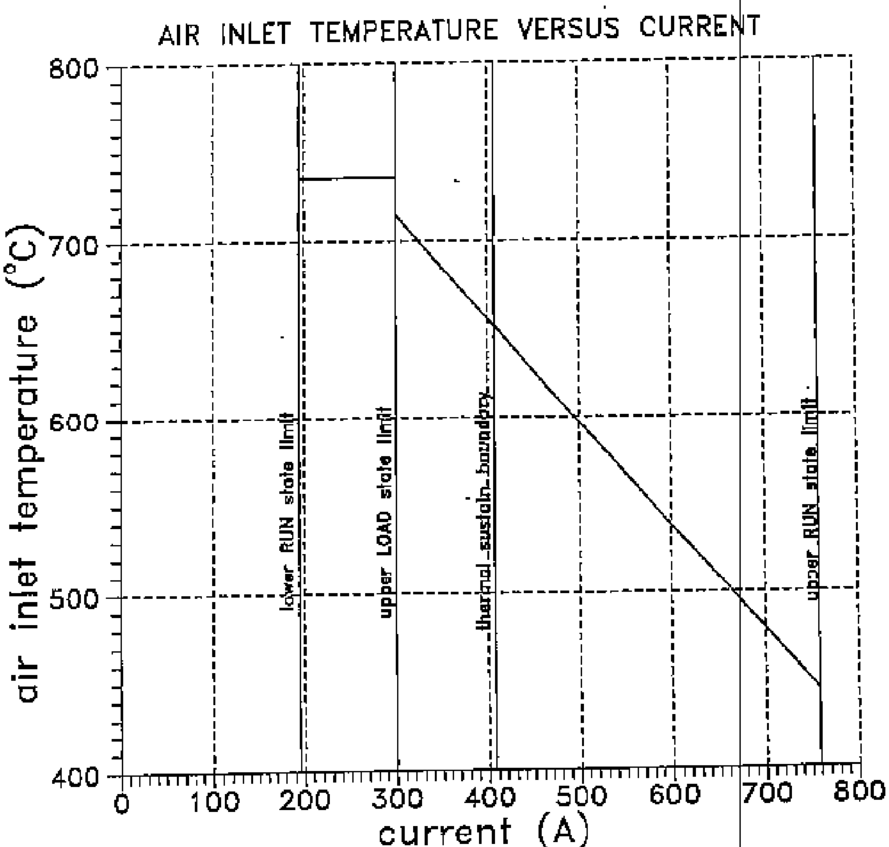

Fig. 14. Air Inlet Temperature Versus Current.

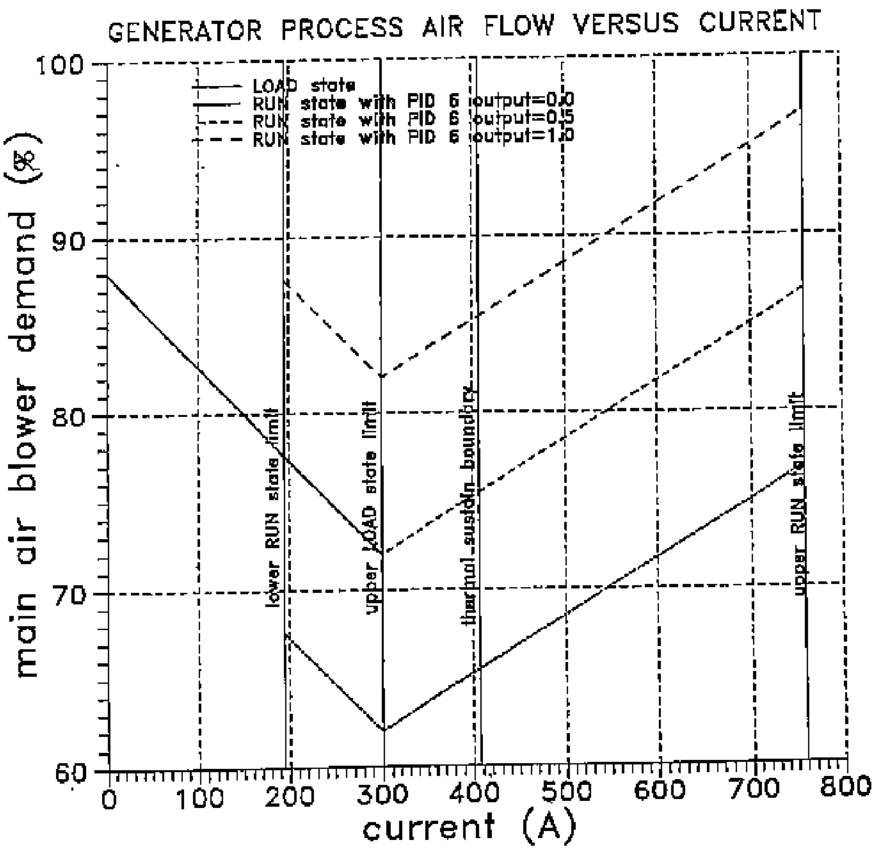

Fig. 15. Generator Process Air Flow Versus Current.

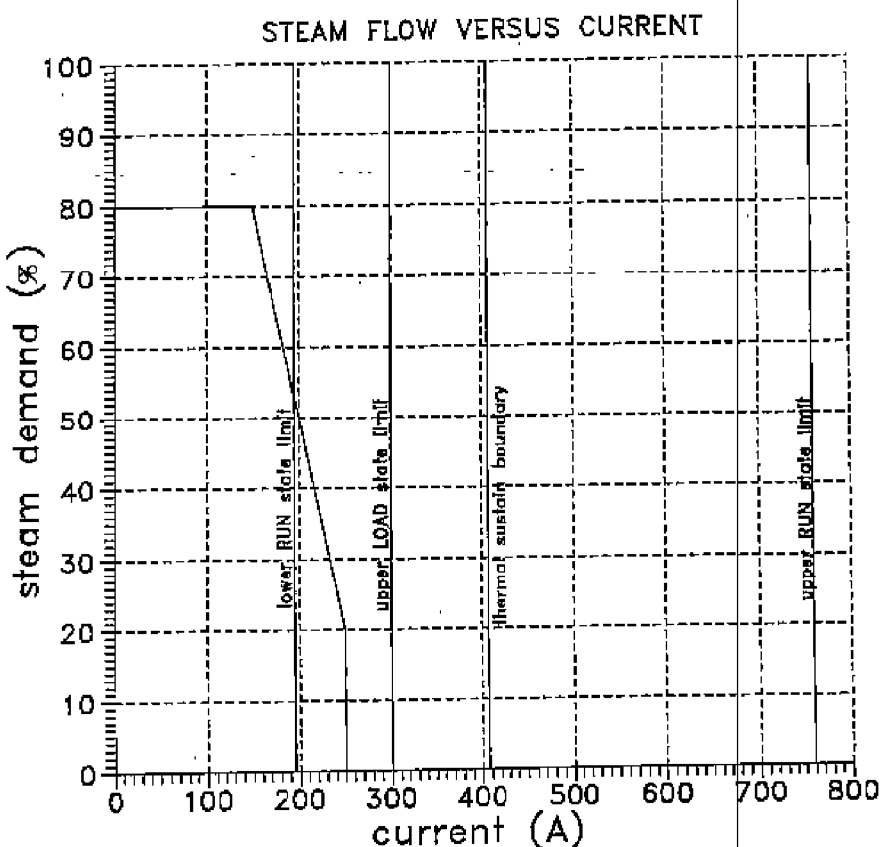

Fig. 16. Steam Flow Versus Current.

operating currents in this state, the steam demand drops linearly with increasing current until it reaches 20% at an operating current of 250 A. For operating currents greater than 250 A, the steam demand vanishes.

Warnings

As shown in the final vertical leg of the flow chart, potential warnings of unallowable conditions are generated from algorithms employing generator temperature, fuel utilization, operational state, and current operator inputs, as well as segment voltages determined previously in the first leg. For example, in the RUN operational state, the allowable generator temperature range is 850 to 1050°C, the allowable current range is 195 to 759 A, and the minimum allowable segment voltage is 8.96 V. While in the LOAD operational state, the minimum allowable generator temperature is 535 °C, the maximum allowable current is 300 A, and the minimum allowable segment voltage is 9.28 V. These values are stored in the file LIMITS.DAT and accessed through the associated NAMELIST assignment in the model. Earlier versions also checked, for warning purposes, whether or not the air flow/fuel flow combination met minimum stoichiometry requirements, depending on the operating current.

Single Operating Point Results

In addition to the parametric exercises already illustrated, the model described above and listed (with associated input files) following the figures was employed to provide single operating point displays of the relevant variables as indicated by the sample in Fig. 17. Inputs were chosen to represent nominal performance in the RUN operational mode at the thermal sustain boundary with EDB/ELSAM fuel.

Conclusion

A FORTRAN language-based model has been developed that can assist in exercising and evaluating selected control options for a 100 kW nominal solid oxide fuel cell generator system. The model has been employed to generate parametric relationships for wide ranges of operating conditions.

REFERENCES

1. Fuel Cells: A Handbook (Revision 3) DOE/METC-94/1006.

2. Training Manual for EDB/ELSAM 100 kW SOFC Generator System, Westinghouse Electric Corporation, Pittsburgh, Pennsylvania

```
fuel = EDB/ELSAM
state = RUN
generator temperature = 1000. degC
fuel utilization = 85.00 %
current = 407.0 A
fuel flow = 386.7 sl/min, 19.23 kg/h
nozzle pressure = .2671 barg
main air blower demand = 75.48 %
current density = 167.5 mA/cm2
cell voltage = .7189 V
segment voltage = 11.50 V
terminal voltage = 276.1 V
gross power = 112.4 kW
air inlet temperature = 651.5 degC
steam demand =    .00 %
methane-based cell electrical efficiencies:
 lower heating value = 48.68 %, higher heating value = 43.87 %
```

Fig. 17. Sample Single Operating Point Display.

FORTRAN Listing of Main SOFC Generator Model

```
C * MODEL FOR 100 kWe SOLID OXIDE FUEL CELL GENERATOR SYSTEM
C    (natural gas flow, air flow, air temperature, current density, cell
C    voltage, segment voltage, stack voltage, gross stack power, nozzle
C    pressure, steam flow)

      implicit real*8(a-h,o-z)
      real*8 lk,lhvch4
      character state*4,fuel*9
      namelist/limits/tgenminl,tgenminr,tgenmaxr,futnminr,futnmaxr,
     1 currmaxl,currminr,currmaxr,vsegminl,vsegminr
      namelist/opinput/fuel,tgen,futncpc,amps,state,pidout

C * read state limit values from file
      open(unit=8,file='limits.dat')
      read(8,limits)

C * read operator input values from file
      open(unit=9,file='opinput.dat')
      read(9,opinput)

C * operator input (generator temperature; not explicitly used here
C    because only available characterization data is for 1000 degC;
C    generally used to control air heaters and/or recuperator bypass
C    valves
c       tgen=1000d0

C * operator input (fuel utilization increment in %)
C    old allowable range:  -3% to +5%
C    new allowable range:  -8% to +5%
c       futncpc=0d0
c       futncpc=2d0
c       futncpc=5d0
c       futilnc=-8d0

C * operator input (current in amperes)
c       ampmin=195d0
c       ampmin=0d0
c       ampmax=759d0
c       ampmax=300d0
c       inc=50
c       ampinc=(ampmax-ampmin)/inc
c       open(unit=7,file='effildel.dat',status='unknown')
c       do 100 i=0,20
c       do 100 i=0,inc
c       amps=510d0
c       amps=1000d0
c       amps=ampmin+i*ampinc
c       if(i.eq.4)amps=300d0
c       if(i.eq.inc)amps=759d0

C * operator input (operating state=HEAT, LOAD, RUN , COOL, or STOP)
c       state='LOAD'
```

```
c * operator input [PID contribution to main air blower demand (to
c    control generator center temperature) can be changed manually by
c    placing PID loop 6 in "manual" and directly setting the output to a
c    value between 0.0 and 1.0; setting the PID loop 6 output to 0.5
c    eliminates any PID loop contribution to blower demand]
c      pidout=0.5d0

c * operator input (air stoichiometry)
c      stoich=5.6d0
c * check to see if specified stoichiometry exceeds minimum requirement
c    for specified current
c      if(amps.le.759d0)then
c       stoichmn=(10.893d0*amps/3d0+14433d0)/(0.01685d0*amps*ncltot/
c     1  ncirc)
c      else
c       stoichmn=3.5d0
c      end if
c      sflag='(SPT)'
c      if(stoich.lt.stoichmn)then
c       sflag='(MIN)'
c       stoich=stoichmn
c      end if

c      pi=3.14159d0
c * specify air composition:  molecular weight, fraction oxygen
c      ema=28.9645d0
c      fro2=0.20948d0

c * specify cell stack arrangement and geometry:  number of circuits,
c    rows in a bundle, bundles in a segment, segments in a pass,
c    passes in the stack, cell length, cell inside diameter, air
c    electrode thickness, electrolyte diameter, electrolyte area
      ncirc=3
      nrwbund=8
      nclbund=ncirc*nrwbund
      nbndseg=2
      nclseg=nclbund*nbndseg
      nsegpas=2
      nclpas=nclseg*nsegpas
      npas=12
      ncltot=nclpas*npas
c      elcl=150d0
c      di=17.8/10d0
c      tae=2.2d0/10d0
c      dlyte=di+tae*2d0
c      aclyte=pi*dlyte*elcl
      aclyte=810d0

c * specify "leak" rate (scc/min)
c      lk=21d0
c    old value
c      lk=18d0
c    new value
      lk=46d0

c * specify fuel type (composition) as EDB/ELSAM, PNG/WESTC, or METHANE
c      fuel='EDB/ELSAM'
```

```
c        fuel='PNG/WESTC'
c * determine average number of moles of H2 and CO from each mole of
c     mixed hydrocarbon fuel
      if(fuel.eq.'EDB/ELSAM')eqhfac=3.5606d0
      if(fuel.eq.'PNG/WESTC')eqhfac=4.056d0
      if(fuel.eq.'METHANE  ')eqhfac=4d0
c * determine molecular weight of fuel
      if(fuel.eq.'EDB/ELSAM')emf=16.5812d0
c      if(fuel.eq.'PNG/WESTC')emf=16.8229d0
      if(fuel.eq.'PNG/WESTC')emf=16.818d0
      if(fuel.eq.'METHANE  ')emf=16d0

c * specify lower and higher heating values for methane (J/standard
c     liter)
      lhvch4=35818d0
      hhvch4=39745d0

c * determine fuel utilization from base and increment
      futbspc=85d0
      futpc=futbspc+futncpc

c * determine current density
      zhay=amps/aclyte*1000d0/3d0

c * use V-J curve fit to 150-cm cell (558) data (ranges:
c     0.49<V<0.75 V, 100<J<480 mA/cm2) for 85% fuel consumption (85% F.U.
c     + 21 sccm) at 1000 degC, 620h, to deduce cell, segment, and stack
c     voltages
      vcel=0.785789d0-2.83202d-4*zhay-6.92928d-7*zhay*zhay
      vseg=nrwbund*nbndseg*vcel
      vterm=ncltot*vcel/3d0

c * checks for/warns of unallowed operating state

c     generator temperature checks
c      LOAD state
      if((state.eq.'LOAD').and.(tgen.le.tgenminl))then
       write(6,10000)tgen,state
10000  format(/,' WARNING:  ',f5.0,' degC generator temperature is too l
     1ow for the ',a4,' state',/)
      end if
c      RUN state
      if((state.eq.'RUN ').and.(tgen.le.tgenminr))then
       write(6,10000)tgen,state
      end if
      if((state.eq.'RUN ').and.(tgen.gt.tgenmaxr))then
       write(6,10000)tgen,state
10004  format(/,' WARNING:  ',f5.0,' degC generator temperature is too h
     1 igh for the ',a4,' state',/)
      end if

c     fuel utilization increment (percent) checks
c      RUN state
      if((state.eq.'RUN ').and.(futncpc.lt.futnminr))then
       write(6,10005)futncpc,state
10005  format(/,' WARNING:  ',f5.0,' % fuel utilization increment is too
     1 small for the ',a4,' state',/)
```

```
      end if
      if((state.eq.'RUN ').and.(futnepc.gt.futnmaxr))then
       write(6,10006)futnepc,state
10006  format(/,' WARNING:  ',f5.0,' % fuel utilization increment is too
     1 large for the ',a4,' state',/)
      end if

c    current checks
c     LOAD state
      if((state.eq.'LOAD').and.(amps.gt.currmaxl))then
       write(6,10003)amps,state
10003  format(/,' WARNING:  ',f4.0,' A current is too high for the ',a4,
     1 ' state',/)
      end if
c     RUN state
      if((state.eq.'RUN ').and.(amps.le.currminr))then
       write(6,10001)amps,state
10001  format(/,' WARNING:  ',f4.0,' A current is too low for the ',a4,
     1 ' state',/)
      end if
      if((state.eq.'RUN ').and.(amps.gt.currmaxr))then
       write(6,10003)amps,state
      end if

c    segment voltage checks
      if((state.eq.'LOAD').and.(vseg.le.vsegminl))then
       write(6,10002)vseg,state
10002  format(/,' WARNING:  ',f4.2,' V segment voltage is too low for th
     1e ',a4,' state',/)
      end if
      if((state.eq.'RUN ').and.(vseg.le.vsegminr))then
       write(6,10002)vseg,state
      end if

c * determine gross stack power
      grpwr=amps*vterm/1000d0

c * old LOAD state startup steam supply control algorithm
c       if(state.eq.'LOAD')then
c        if(amps.gt.150d0)then
c         stdm=-0.0433d0*amps+13.7d0
c        else
c         stdm=7.2d0
c        end if
c       end if

c * new LOAD state startup steam supply control algorithm
      if(state.eq.'LOAD')then
       if(amps.le.150d0)then
        stdmpc=80d0
       else if((amps.gt.150d0).and.(amps.le.250d0))then
        stdmpc=-0.6d0*amps+170d0
       else
        stdmpc=0d0
       end if
      end if
```

```
c * determine air flow set point
c      asp=0d0
c * new HEAT state process air supply control algorithm
      if(state.eq.'HEAT')bldmpc=68d0
c * new LOAD state process air supply control algorithm
      if(state.eq.'LOAD')bldmpc=-0.0533d0*amps+88d0
c * new RUN state process air supply control algorithm
      if(state.eq.'RUN ')then
       if(amps.le.300d0)then
        bldmpc=-0.0533d0*amps+88d0+(20d0*pidout-10d0)
       else
        bldmpc=0.0325d0*amps+62.25d0+(20d0*pidout-10d0)
       end if
      end if
c * new COOL state process air supply control algorithm
      if(state.eq.'COOL')bldmpc=88d0
c * new STOP state process air supply control algorithm
      if(state.eq.'STOP')then
       if((amps.lt.10d0).and.(vterm.gt.290d0))then
        bldmpc=6d0
       else
        bldmpc=67d0
       endif
      endif

c * old LOAD state nitrogen/hydrogen mix control algorithm??
c    may be tough because flow is determined by relation between
c    purge gas supply pressure and induced fuel nozzle backpressure
c    (p. 199)

c * determine fuel flow set point
c * determine fuel standard density (grams/standard liter)
      cgk=emf/22.414d0
c      gsp=0d0
c       gsp=2.5d0+amps*(13.4d0-2.5d0)/300d0
c * HEAT state natural gas supply algorithm
      if(state.eq.'HEAT')gsp=0d0
c * new LOAD state natural gas supply algorithm (modified-factor of 10)
      if(state.eq.'LOAD')then
       gsp=(0.799d0*amps+50d0)*0.4971d0/10d0
       gslpm=gsp*1d3/cgk/60d0
      end if
      if(state.eq.'RUN ')then
c * determine the required reacted fuel (H2 or CO) for each cell per amp
c    generated:  (coulombs/second)*(electrons/coulomb)*
c    (standard liters/mole)*(seconds/minute)/(electrons/molecule)/
c    (molecules/mole)=standard liters/min
       cgsl=6.242d18*22.414d0*60d0/2d0/6.023d23
c * old RUN state natural gas supply algoritm
c        gslpm=(cgsl*amps/ncirc+1k/1d3)*ncltot/eqhfac/fc
c * new RUN state natural gas supply algorithm
       gslpm=(cgsl*amps/ncirc+1k/1d3)*ncltot/0.976d0/eqhfac/(futpc/1d2)
c * determine fuel setpoint (kilograms/hour)
       gsp=gslpm*cgk*60d0/1d3
      end if
c * COOL state natural gas supply algorithm
      if(state.eq.'COOL')gsp=0d0
```

```
c * STOP state natural gas supply algorithm
      if(state.eq.'STOP')gsp=0d0

c * nozzle pressure
      pnoz2=(gsp-6.423d0)/47.96d0

c * determine methane-based low and high heating value cell electrical
c    efficiencies
      efflhv=100d0*grpwr*1000d0*60d0/lhvch4/gslpm
      effhhv=100d0*grpwr*1000d0*60d0/hhvch4/gslpm

c        casl=cgsl/2d0/fro2
c        afcslpm=stoich*casl*amps/ncirc*ncltot
c        cak=60d0*ema/22.414d0/1d3
c        afc=afcslpm*cak
c        ald=40d0
c        asp=afc+ald
c        asp=stoich*0.0167d0*amps*ncltot/ncirc*cak

c * LOAD state air inlet temperature control algorithm
      if(state.eq.'LOAD')then
       tairin=735d0
      end if

c * new RUN state air inlet temperature control algorithm
      if(state.eq.'RUN ')then
       if(amps.gt.300d0)then
        tairin=-0.5885d0*amps+891.0d0
       else
        tairin=735d0
       end if
      end if

c      write(6,4000)amps,gsp,bldmpc,tairin,pnoz2
c      write(7,4000)amps,gsp,bldmpc,tairin,pnoz2
c      write(6,7000)amps,vterm,grpwr,vseg
c      write(7,7000)amps,vterm,grpwr,vseg
 7000 format(4f10.3)
 4000 format(5f10.3)
      write(6,1000)fuel,state
c      write(6,2000)tgen,futpc,stoich,sflag
      write(6,2000)tgen,futpc,amps
c      write(6,3000)amps,asp,gslpm,gsp
      write(6,3000)gslpm,gsp
      write(6,8000)pnoz2
      write(6,3500)bldmpc
      write(6,4500)zhay
      write(6,5000)vcel,vseg,vterm
      write(6,6000)grpwr
      write(6,8100)tairin,stdmpc
      write(6,8200)efflhv,effhhv
c      write(7,20000)amps,stdmpc
c      write(7,30000)amps,efflhv,effhhv
30000 format(f7.2,2f6.2)
20000 format(f5.0,f6.2)
  100 continue
 1000 format(' fuel = ',a9,/,' state = ',a4)
```

```
c 2000 format(' generator temperature=',f5.0,'degC; fuel utilization=',
c      1 f4.1,'%; air stoichs=',f4.2,a5)
 2000 format(' generator temperature = ',f5.0,' degC',/,
     1 ' fuel utilization = ',f5.2,' %',/,' current = ',f5.1,' A')
c 3000 format(' current=',f5.1,'amps; air flow=',f5.0,'kg/h; fuel flow=',
c      1 f6.2,'sl/min, ',f5.2,'kg/h')
 3000 format(' fuel flow = ',f5.1,' sl/min, ',f5.2,' kg/h')
 3500 format(' main air blower demand = ',f5.2,' %')
 4500 format(' current density = ',f5.1,' mA/cm2')
 5000 format(' cell voltage = ',f5.4,' V',/,' segment voltage = ',f5.2,
     1 ' V',/,' terminal voltage = ',f5.1,' V')
 6000 format(' gross power = ',f5.1,' kW')
 8000 format(' nozzle pressure = ',f5.4,' barg')
 8100 format(' air inlet temperature = ',f5.1,' degC',/,
     1 ' steam demand = ',f5.2,' %')
 8200 format(' methane-based cell electrical efficiencies:',/,'  lower h
     1eating value = ',f5.2,' %, higher heating value = ',f5.2,' %')
      end
```

LIMITS.DAT File for Use by Main SOFC Generator Model

```
&limits
 tgenminl=535d0,
 tgenminr=850d0,
 tgenmaxr=1020d0,
 futnminr=-8d0,
 futnmaxr=5d0,
 currmaxl=300d0,
 currminr=195d0,
 currmaxr=759d0,
 vsegminl=9.28d0,
 vsegminr=8.96d0
/
```

OPINPUT.DAT File for Use by Main SOFC Generator Model

```
&opinput
 fuel='EDB/ELbAM',
 tgen=1000d0,
 futncpc=0d0,
 amps=407d0,
 state='RUN ',
 pidout=0.5d0
/
```